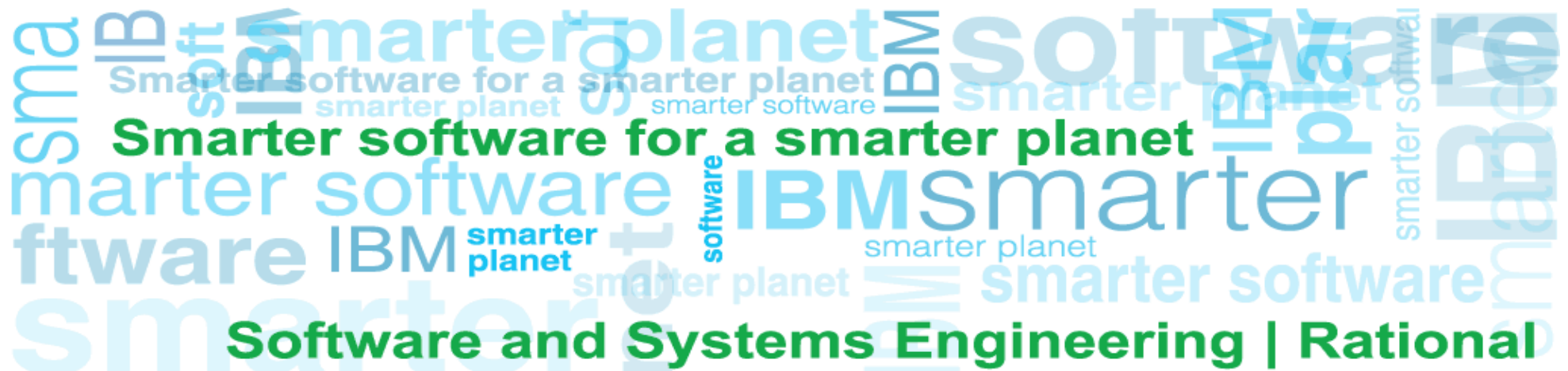


Practical Agile Approach

Taemin Ko (고대민 차장)
Application Consultant, IBM Rational Services
tmk@kr.ibm.com



Agenda

- Overview
- DAD, Disciplined Agile Delivery
- IBM Agile Services
- IBM Case Study: ClearCase with Agile
- Summary

Overview

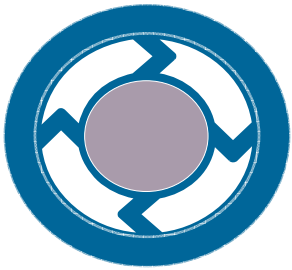
The presentation is to introduce IBM's Practical Agile Approach

- DAD, Disciplined Agile Delivery
 - *for agile adoption in real world, various development environment*
- IBM Agile Services
 - *for customized agile adoption*
- IBM's internal case study
 - *Globally distributed development*
 - *Transition from traditional development to agile adoption*
- IBM's Insights on Agile adoption

Agenda

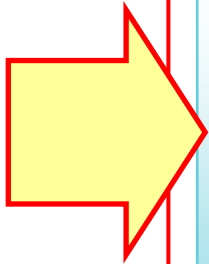
- Overview
- DAD, Disciplined Agile Delivery
- IBM Agile Services
- IBM Case Study: ClearCase with Agile
- Summary

Agile Scaling Model (ASM)



Core Agile Development

- Focus is on construction
- Goal is to develop a high-quality system in an evolutionary, collaborative, and self-organizing manner
- Value-driven lifecycle with regular production of working software
- Small, co-located team developing straightforward software



Disciplined Agile Delivery

- Extends agile development to address full system lifecycle
- Risk and value-driven lifecycle
- Self organization within an appropriate governance framework
- Small, co-located team delivering a straightforward solution

Agility at Scale

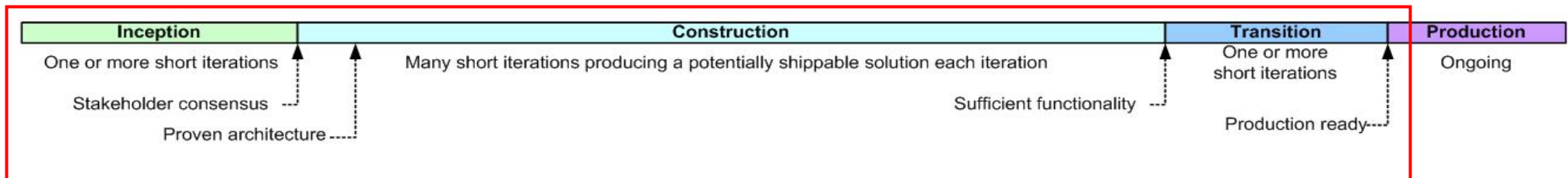
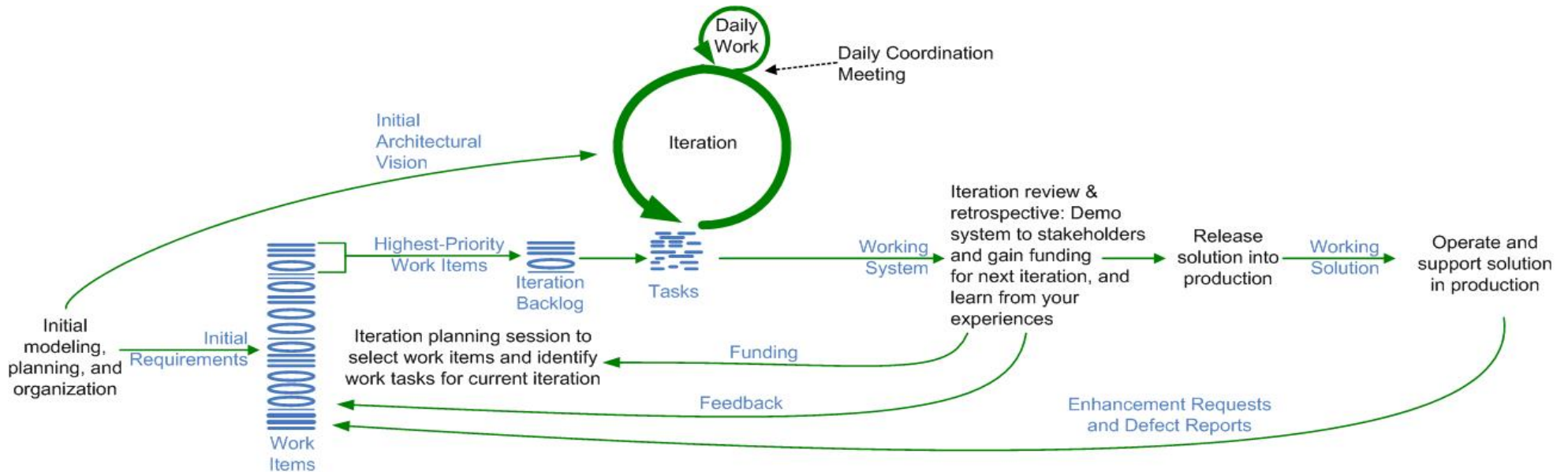
- Disciplined agile delivery and one or more scaling factors applies

What is disciplined agile?

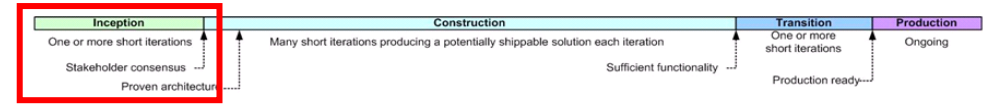
- Disciplined agile delivery is an evolutionary (iterative and incremental) approach that regularly produces high quality solutions in a cost-effective and timely manner via a risk and value driven lifecycle.
- It is performed in a highly collaborative, disciplined, and self-organizing manner within an appropriate governance framework, with active stakeholder participation to ensure that the team understands and addresses the changing needs of its stakeholders.
- Disciplined agile delivery teams provide repeatable results by adopting just the right amount of ceremony for the situation which they face.



The disciplined agile lifecycle: An extension of Scrum

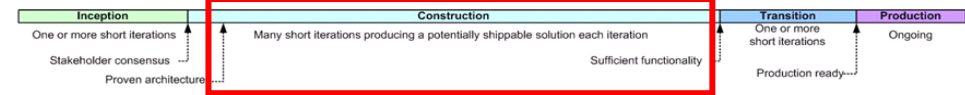


Inception

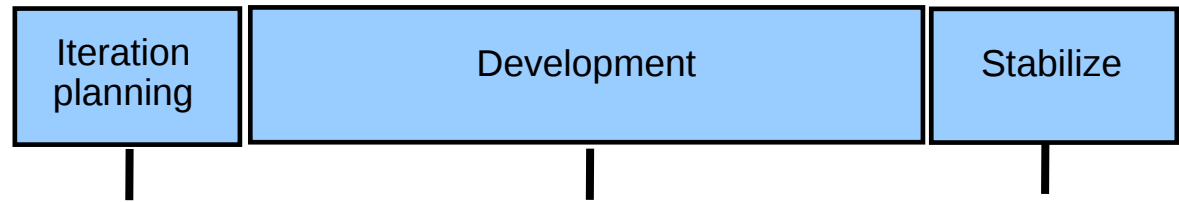


- Initiation activity to ensure that the project gets off to a good start
- Initiation activities include:
 - Initial requirements envisioning
 - Initial architecture envisioning
 - Initial planning
 - Putting the team together
 - Setting up your work environment
 - Gaining stakeholder consensus regarding the scope and plan

Construction



- Many short iterations⁽¹⁾ producing a potentially shippable solution each iteration
- Agile teams work collaboratively
- Agile teams are self organizing; they plan and estimate the work that they do
- Systems are developed incrementally, with each iteration adding new or improved functionality
- Iterations have rhythms:



Organize the work to be done during this iteration

Do the work

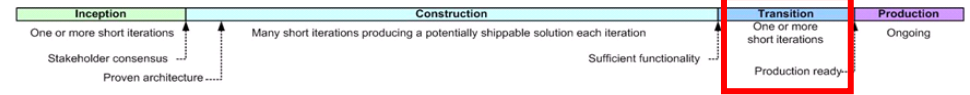
- Ensure that your system is “potentially shippable”
- Demonstrate the system
- Hold a reflection session

Practices

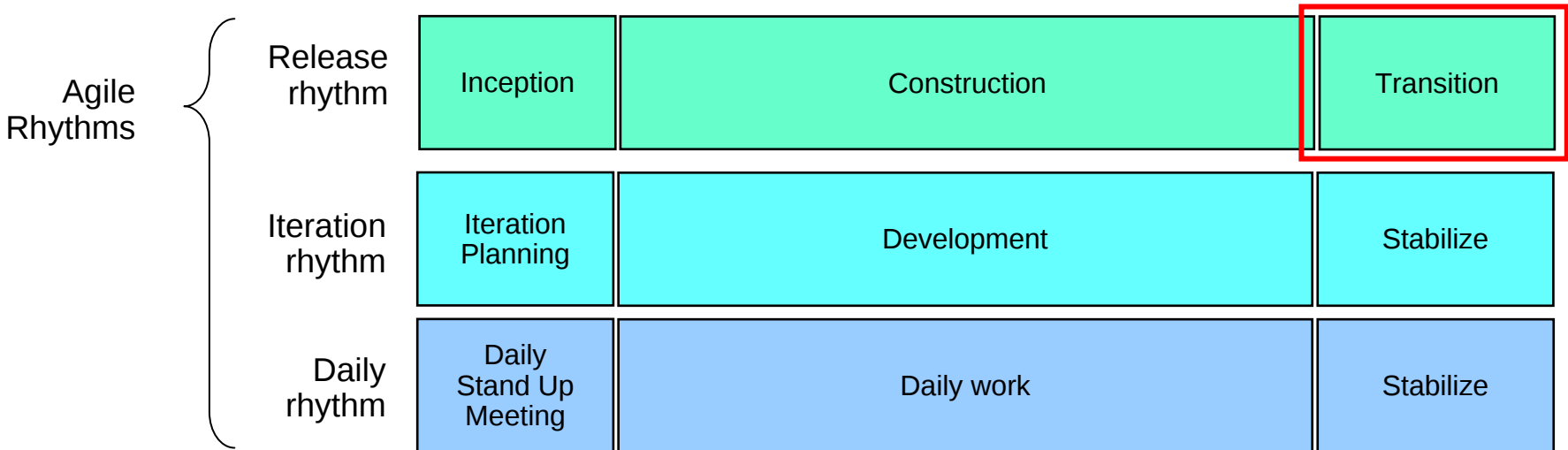
- Iteration Planning
- Architecture envisioning (high-level design)
- Test-driven development (TDD)
- Refactoring
- ...

(1) An iteration is a time-boxed milestone that is used to measure a project's progress based on working software increments

Transition



- How is “Agile transition” different from “traditional transition” ?
 - In many ways agile transition is similar to traditional approaches to transition
 - Beta/pilot the solution, End-of-lifecycle testing, Finalize documentation, Stabilize the code
 - However, agile approaches:
 - Require less end-of-lifecycle testing, less effort to stabilize the code, light-weight documentation
 - Require less time to transition due to the greater focus on quality throughout the life cycle
- Agile approaches require less end-of-lifecycle testing than traditional approaches, due to their greater focus on testing, but still require these activities



Agenda

- Overview
- DAD, Disciplined Agile Delivery
- IBM Agile Services
- IBM Case Study: ClearCase with Agile
- Summary

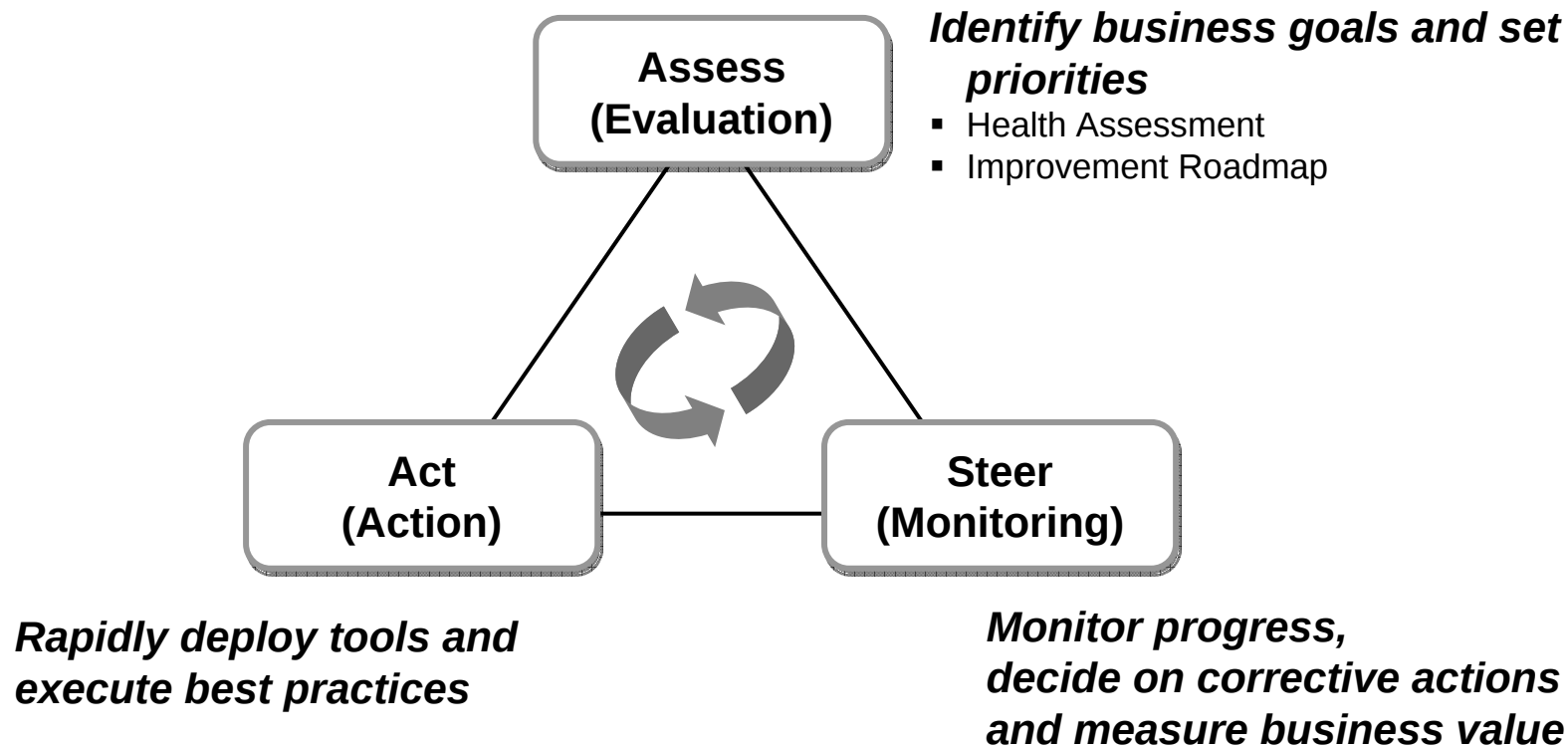
IBM Agile Services

- Our Measured Capability Improvement Framework (MCIF) service offering helps organizations to successfully improve their IT practices in a sustained manner
- Agile mentoring and coaching:
 - We understand the enterprise-level issues that you face
 - We scale from pilot project consulting to full-scale agile adoption
- Agile training:
 - Disciplined Agile Delivery (DAD) workshops
 - Agile and Rational Team Concert (RTC)
 - Customized training
- Our Accelerated Solutions Delivery (ASD) practice has years of experience delivering agile projects at scale



MCIF⁽¹⁾

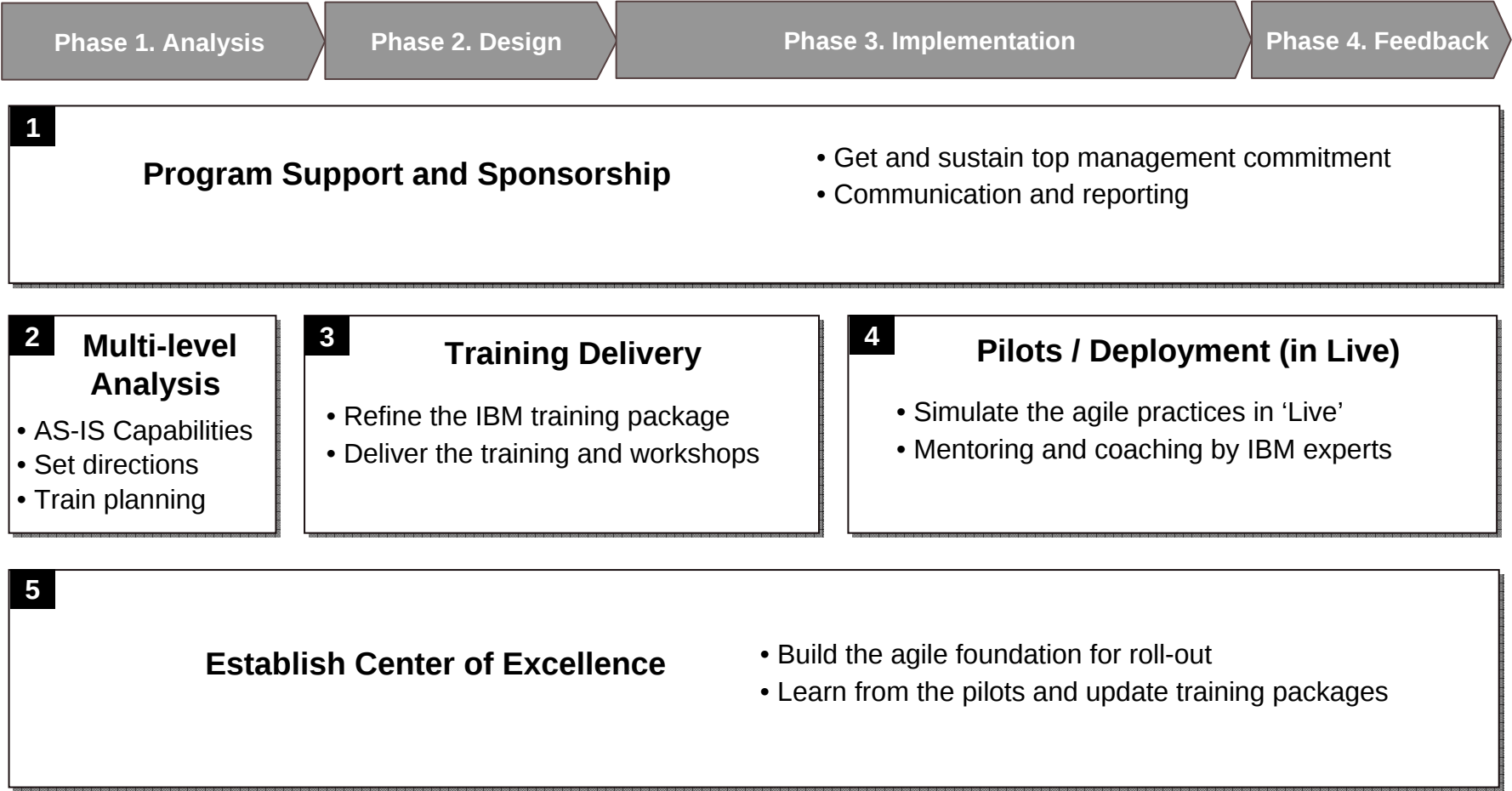
MCIF is IBM Rational's assessment and improvement framework that applies Rational capabilities, best practices and services to improve software and systems delivery in iterative, continuous and measurable manners.



(1) MCIF: Measurable Capability Improvement Framework

Mentoring & Coaching > Overview

Mentoring & Coaching applies from pilot project to full-scale agile adoption with agile principles, practices, customer understanding, practical learning and continuous feedback



Mentoring & Coaching > Multi-level Analysis

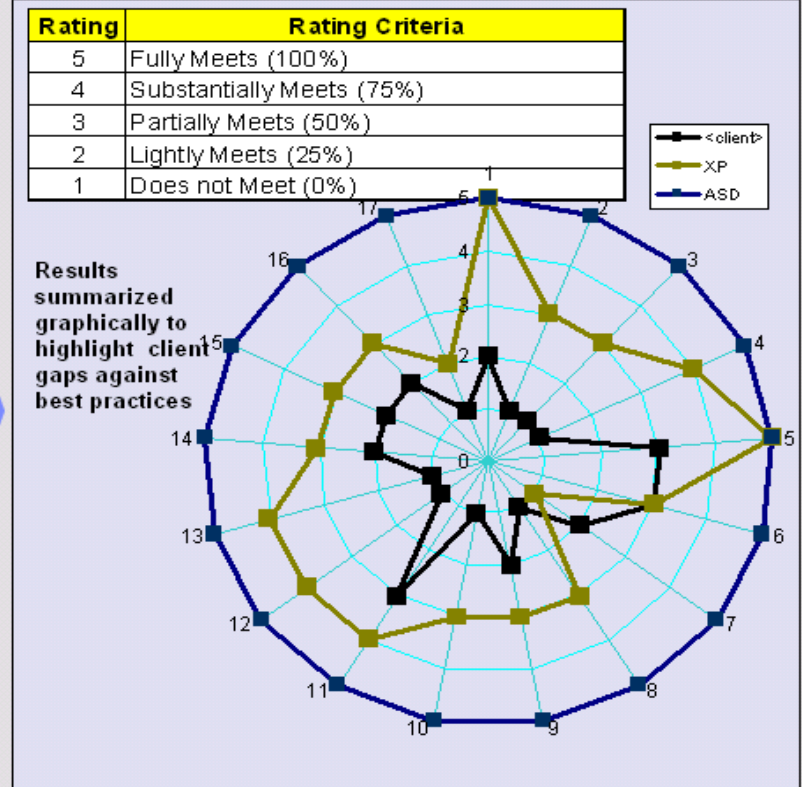
Quickly identify barriers to agility, and end to end areas for improvement, that are then used as inputs to project level or organization level optimization plans *



Three weeks or less:

- Interview project team(s) and business / IT stakeholders
- Summarize findings and apply rating system
- Develop proposed actions to resolve gaps
- Present findings to client sponsors
- Jointly draft plans to implement actions (short & long term)

Category	Number	Organization Level Practices	Rating	
			<client>	ASD
Process	1	Accelerators are planned for and used	1	5
	2	Strong focus on project management (art and science)	1	5
Org	Project Level Accelerators		Rating	
	Number		<client>	ASD
	1	Onsite customer / Co-location with project team	1	5
	2	Collaborative Planning	1	5
	3	Small releases / Timeboxing	1	5
	4	Test First / Testing Specialist and tools	1	5
People / Skills	5	Refactoring	1	5
	6	Peer programmer assistance (10 minute rule)	1	5
	7	Coding standards	1	5
Metrics	8	40 hour week	1	5
	9	Scope prioritization	1	5
	10	Iterative analysis, prototyping/design & development	1	5
Technology	11	Facilitated Joint Application Requirements & Design	1	5
	12	Reuse when it makes business sense	1	5
	13	Agile (but scalable) processes and documentation	1	5
	14	Small, dedicated project teams (4 to 8)	1	5
	15	Dedicated facilities and equipment	1	5
	16	development exists and is consistently used		
	17	Client strategic products and architectures are iteratively enabled through project delivery	1	5



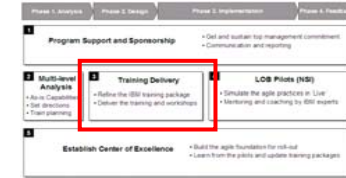
Value to Client

- Low cost approach to quickly identify areas for improvement
- Opportunity to discuss ideas for resolution of gaps with IBM experts

** Can also be integrated with other Application Development Effectiveness assessment components*

Agile Training

Designed on the practical, innovative principles adapting various methods



The Program Design Principles

1 Action-oriented

- Provide participants with first-hand experiences applicable to the real world business through case studies and hands-on exercises

2 Focused on Capability Building

- Focus on building capabilities of each individuals to apply concepts and frameworks learned to the their day-to-day jobs by providing insights and thinking process

3 Enabling Out-of-the Box Thinking

- Provide opportunities to exercise the concepts and methods in the real world projects with outstanding and experience IBM agile experts in order to learn 'live' expertise

4 Broadening Perspectives

- Broaden perspectives of customer through discussions with world renowned thought leaders in various subject areas; thus, enabling them to rethink software delivery in the context of the development strategy and customer impact

Applied Methods

Components

- Lectures
- Case Studies
- Exercises

Pilot

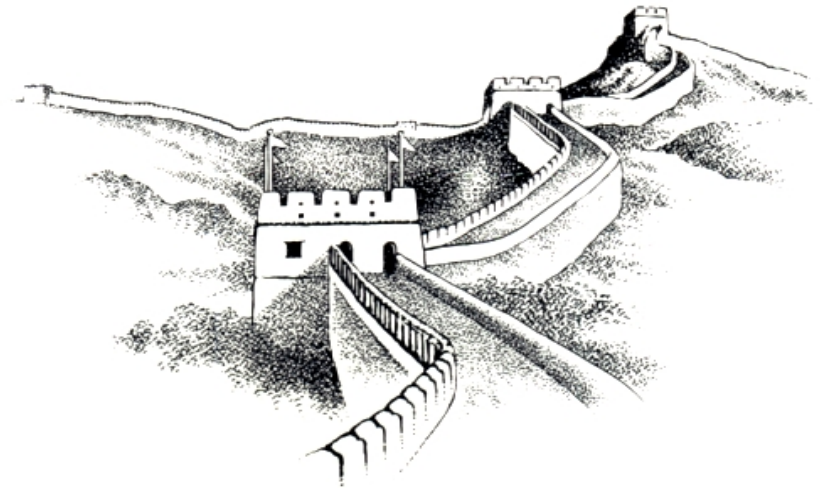
- Can go and exercise agile theory in real-world examples in person
- Can share ideas and opinions with participating IBM experts and coaches, and evangelist

Sample Course Offerings

- Agile Practices Workshop
- Disciplined Agile Delivery
- Leading Agile Teams
- Agile overview
- Agile Automation Environment

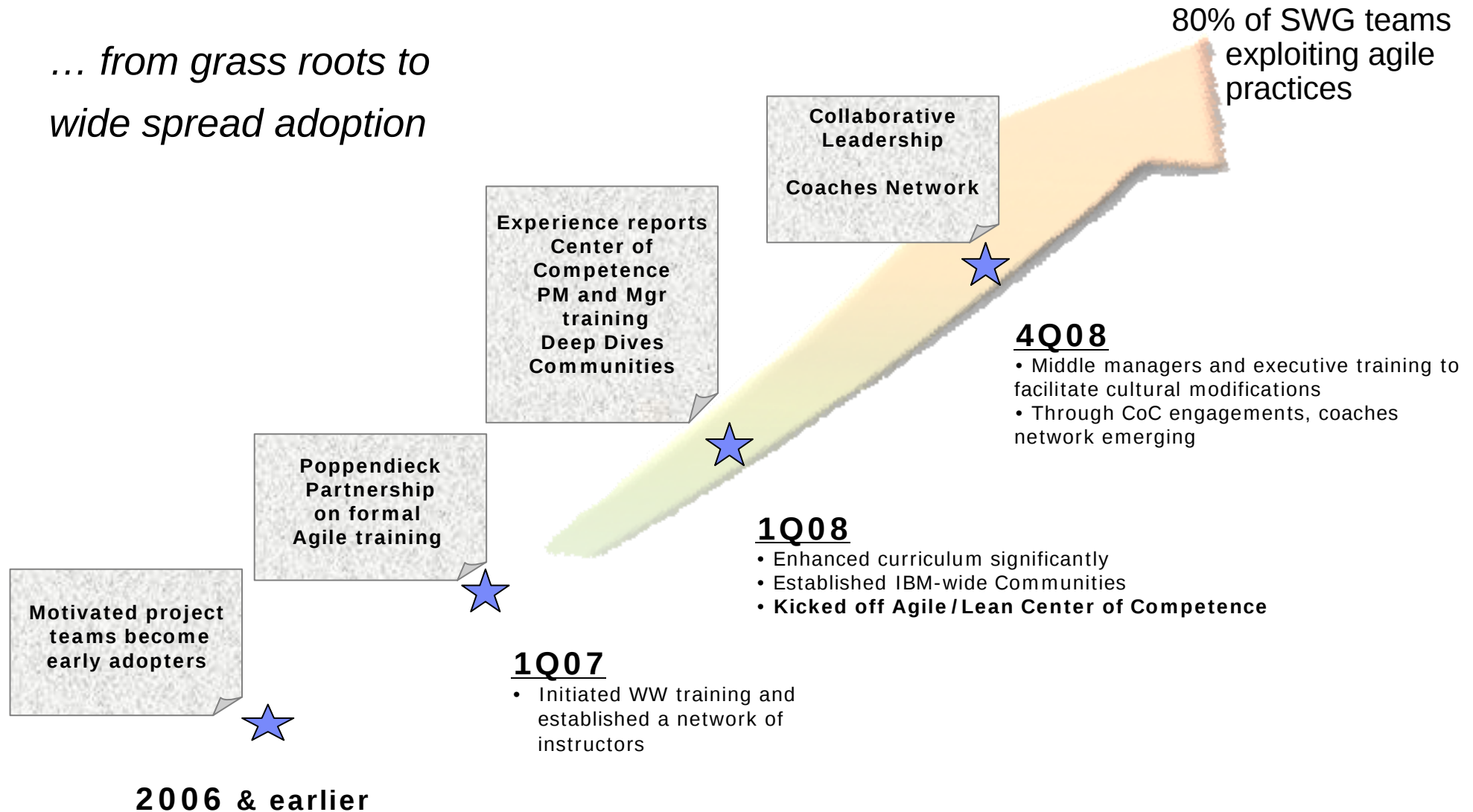
Agenda

- Overview
- DAD, Disciplined Agile Delivery
- IBM Agile Services
- IBM Case Study: ClearCase with Agile
- Summary



Agile adoption in IBM Software Group (SWG)

... from grass roots to wide spread adoption



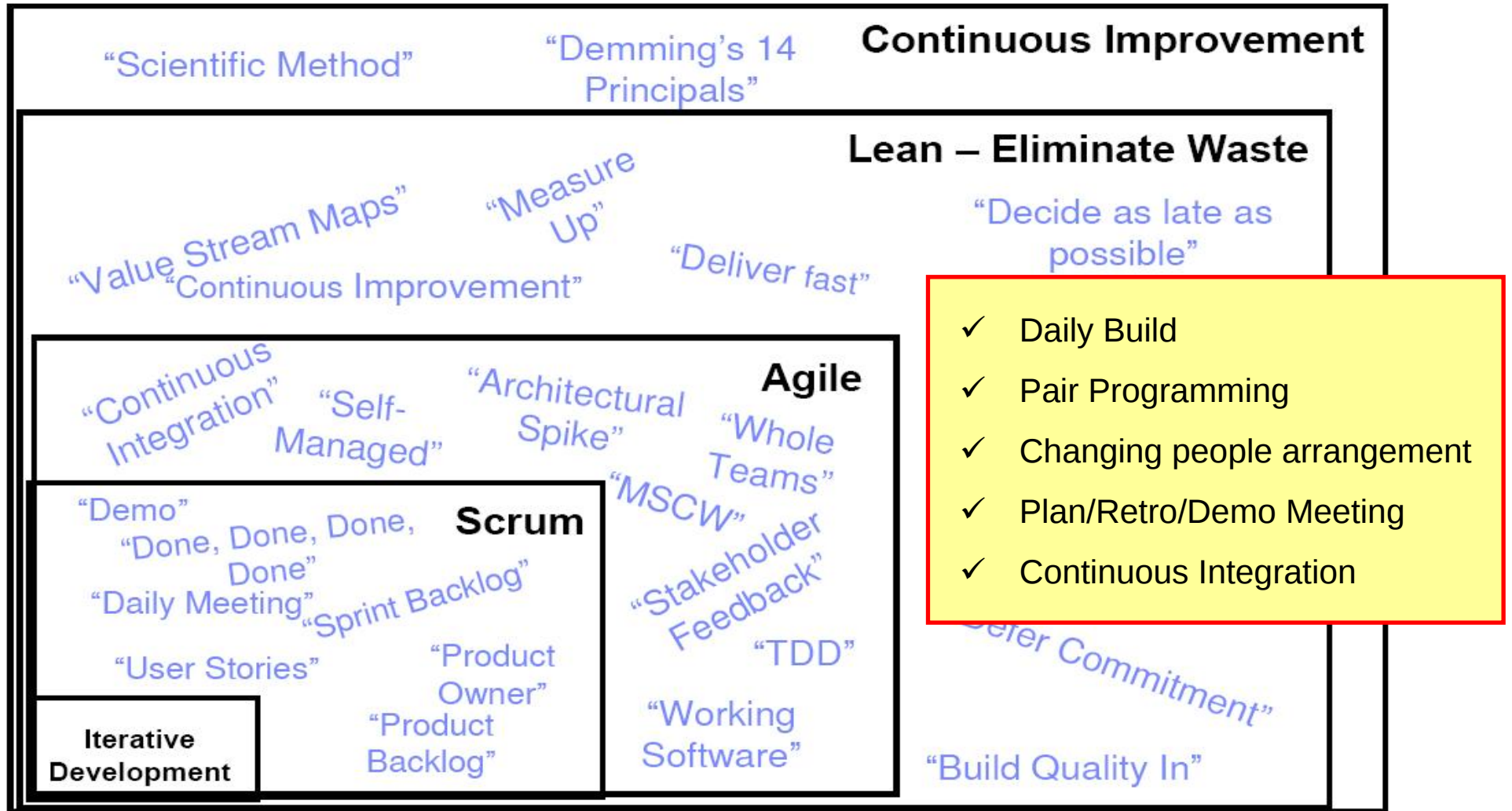
Project Status

Background	▪ Project brief introduction – Core (CC Core and CQ Core) – Client (CCRC, CC Native, CQ Web, CQ Native) – CM Server (Support CC and CQ) ▪ Team – Transferred from a testing team : CDL – Traditional project management – Globally distributed in China and US
------------	---

Pain Points
▪ Feel Pain in communication ▪ Slow learning progress ▪ Unpredictable progress ▪ Low productivity ▪ Low morale

Problem Analysis
▪ Dealing with the distributed teams – 12 Hours Time Different • Hard to find a person to ask question • Hard to find a time for discussion ▪ Ineffective Trainings and Meetings – One way communication – No face to face talk – Training method is ineffective ▪ Unbalancing Technical Experience – Fresh Members V.S. Experience Members – Junior V.S. Senior ▪ Low Morale – Not self-management – Lack of initiative

The methodologies choose from...



1st Step @ Agile (1)

Daily Scrum

Pair Programming

Challenges

- Distributed Team

- Ineffective Trainings and Meetings
- Unbalance Technical Experience

How to did

- Make it Short
- High frequency
- Focus on blocking issues
- Fully use of Conference Tools
 - Lotus Web Conference
 - Net Meeting

- Leverage communication tools
- Choose proper task for pairing
- Be very patient and industrious

1st Step @ Agile (2)

Changing people arrangement

Plan/Retro/Demo Meeting

Challenges

- Unbalance Technical Experience
- Making the communication effective
- Top challenge: Low Morale

How to did

- Form layer based team
 - Break down CDL team
 - Mix US members and CDL members
 - Form Scrum Team based on Layer
 - Core
 - CM Server
 - Client
- Collect every member's complains and suggestions
- Example

1st Step @ Agile (3)

Continuous Integration

Challenges

- Getting to done within a sprint

Nightly Build

How to did

- Build Master Process
- Leverage Tools
 - Build forge

The screenshot displays the Rational Build Forge web interface. The top navigation bar includes 'Rational Build Forge' and 'IBM'. The main content area shows a project named 'WebAuthBuild' with a selector 'ChakPC' and environment 'Env: --'. The build process is listed with 11 steps, each with a status icon (green for success, red for failure) and a result. The steps are: 1. BuildDevEAR, 2. JUnitTestCases, 3. PredeployPredev, 4. VerityParamPredev, 5. DeployPredev, 6. RestartPredevServer, 7. VerityLogin, 8. DeployDev, 9. PredeployFVT, 10. DeployFVT, and 11. RestartFVTServer. All steps show a green status icon and a result of 'AntBuildWebAuth'. The bottom section shows the details for the selected step 'BuildDevEAR'.

#	Step Name	Selector	Environment	Result	Access
1	BuildDevEAR			AntBuildWebAuth	Default
2	JUnitTestCases			AntBuildWebAuth	Default
3	PredeployPredev			AntBuildWebAuth	Default
4	VerityParamPredev			AntBuildWebAuth	Default
5	DeployPredev			AntBuildWebAuth	Default
6	RestartPredevServer			AntBuildWebAuth	Default
7	VerityLogin			AntBuildWebAuth	Default
8	DeployDev			AntBuildWebAuth	Default
9	PredeployFVT			AntBuildWebAuth	Default
10	DeployFVT			AntBuildWebAuth	Default
11	RestartFVTServer			AntBuildWebAuth	Default

Are we Agile after first step?
(Do retrospect for our first step)

Our Agile Definition: Uses continuous stakeholder feedback to deliver high-quality, consumable code through use cases and a series of short, stable, time-boxed iterations.

- Stakeholder feedback
- High-quality, Consumable code
- Use cases (stories)
- Time-boxed iteration

2nd Step @ Agile (1)

Changed People organization again

Make some part more fun

Challenges

- Ineffective communication while bug fixing
- Getting to done within a sprint
- Dealing with the distributed teams
- Low Morale
- More self Management

How to did

- Characteristic Scrum Team
 - Layer based for new features development
 - Component scrum team for bug fixing
- Tasks: Assigned by Lead
 - Self select
- Priority: Assigned by Lead
 - Planning Poker (velocity)

2nd Step @ Agile (2)

Make part of rules easy to follow

Customized Several Tools

Challenges

- Low Morale

- Productivity

How to did

- Daily scrum
→ Twice a week scrum
+ daily scrum report
- Reflection become not mandatory

- Customize handy tool RBB
 - RTC
 - CC Bridge
 - CQ Bridge

Agenda

- Overview
- DAD, Disciplined Agile Delivery
- IBM Agile Services
- IBM Case Study: ClearCase with Agile
- Summary

Summary

Agile adoption
in real world

- Agile principles and practices needs scaling for full lifecycle and enterprise development environments
- DAD, Disciplined Agile Delivery and Agile@Scale for agile adoption in real world

IBM Agile Services

- MCIF (Measured Capability Improvement Framework) with Practices
- Customized Agile Mentoring & Coaching, Agile Training
- Accelerated Solutions Delivery (ASD) with years of experience delivering agile projects at scale

Key success
factors at
Agile@IBM

- Process, People and Tool
- Iterative, continuous and measurable efforts
- Consistent awareness and involvement from top managers to practitioners



Taemin Ko (고대민 차장)
Application Consultant, IBM Rational Services
tmk@kr.ibm.com

© Copyright IBM Corporation 2011. All rights reserved.

The information contained in these materials is provided for informational purposes only, and is provided AS IS without warranty of any kind, express or implied. IBM shall not be responsible for any damages arising out of the use of, or otherwise related to, these materials. Nothing contained in these materials is intended to, nor shall have the effect of, creating any warranties or representations from IBM or its suppliers or licensors, or altering the terms and conditions of the applicable license agreement governing the use of IBM software. References in these materials to IBM products, programs, or services do not imply that they will be available in all countries in which IBM operates. Product release dates and/or capabilities referenced in these materials may change at any time at IBM's sole discretion based on market opportunities or other factors, and are not intended to be a commitment to future product or feature availability in any way.

IBM, the IBM logo, the on-demand business logo, Rational, the Rational logo, and other IBM products and services are trademarks of the International Business Machines Corporation, in the United States, other countries or both. Other company, product, or service names may be trademarks or service marks of others.